Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Sketch Based Image Retrieval for Architecture Images with Siamese Swin Transformer

To cite this article: Yuxin Xu *et al* 2022 *J. Phys.: Conf. Ser.* **2278** 012035

View the article online for updates and enhancements.

You may also like

- Siamese Network with multi-scale fusion attention for Visual Tracking
Xue Shangjie, Yao Wenjin and Yang Wenjun
- Multi^3: *multi*-templates siamese network with *multi*-peaks detection and *multi*-features refinement for target tracking in ultrasound image sequences
Yifan Wang, Tianyu Fu, Yan Wang et al.
- Learning image representations for content-based image retrieval of radiotherapy treatment plans
Charles Huang, Varun Vasudevan, Oscar Pastor-Serrano et al.

This content was downloaded from IP address 95.61.64.225 on 05/03/2024 at 08:48

Sketch Based Image Retrieval for Architecture Images with Siamese Swin Transformer

Yuxin Xu[1,a], Yuyao Yan[2,b], Yiming Lin[1,c], Xi Yang[1,d], Kaizhu Huang[1,*,e]

[1]Xi'an Jiaotong-Liverpool University, 111 Ren'ai Road, Suzhou Industrial Park, Suzhou, Jiangsu Province, P. R. China.
[2]Suzhou Bitspring Technology Co., Limited, Jiangsu Province, P. R. China.

E-mail: {[a]yuxin.xu20, [c]yiming.lin21}@student.xjtlu.edu.cn, {[d]xi.yang01, [*,e]kaizhu.huang}@xjtlu.edu.cn, [b]yuyao.yan@0q.design

Abstract. Sketch-based image retrieval (SBIR) is an image retrieval task that takes a sketch as input and outputs colour images matching the sketch. Most recent SBIR methods utilise deep learning methods with complicated network designs, which are resource-intensive for practical use. This paper proposes a novel compact framework that takes the siamese network with image view angle information, targeting the SBIR task for architecture images. In particular, the proposed siamese network engages a compact SwinTiny transformer as the backbone encoder. View angle information of the architecture image is fed to the model to further improve search accuracy. To cope with the insufficient sketches issue, simulated building sketches are used in training, which are generated by a pre-trained edge extractor. Experiments show that our model achieves 0.859 top-one accuracy exceeding many baseline models for an architecture retrieval task.

1. Introduction

Image retrieval is a challenging task in computer vision aiming to search and retrieve images in a given image database. Content-based image retrieval (CBIR) and image-to-image retrieval are two classic image retrieval methods. They apply a semantic description or a reference image as the input, and retrieve relevant images. Both image retrieval methods are used widely in conventional search engines. Despite being a less popular image retrieval method than CBIR and image-to-image retrieval, Sketch-based image retrieval (SBIR) is a well-researched topic as well. It is more suitable than the other two methods in searching images that are hard to describe by words, or when no reference image is available as input. One example of such a task is finding building designs with a sketch. Architects may want to gain inspiration from existing buildings when creating new designs. However, buildings are hard to differentiate by simple language since they are abstract designs created by architects. With an SBIR model, architects could identify similar buildings by providing a single sketched image. Existing SBIR methods with high precisions would yield ideal results on this task. However, high-accuracy models are often resource-intensive, making them hard to apply in practical cases. To resolve this problem, we propose a compact SBIR model dedicated to building and architecture images. Fig. 1 shows two sample sketches and their top matches produced by the proposed model. The majority of SBIR models can be categorised into feature descriptor models and deep learning models. The idea of feature descriptor models, such as Bag-of-Features Descriptors [1], is to

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

extract specific features (e.g. edge feature, HOG feature) from the inputs and compare them to each other for similarities using an indexing method or a classification method, such as a state vector machine [2].

Figure 1. Results for the proposed model: top-six matches for two hand-drawn sketches [3] retrieved from a building image pool with 34,226 images collected from ArchDaily [4].

Deep learning models have outstanding performance on computer vision tasks [5], SBIR is no exception. Deep learning models rely on a data-driven approach to automatically learn the domain gap between images and sketches, making them more robust compared to hand-crafted feature extractors. Classic deep learning models developed for SBIR tasks exploit siamese networks with CNN backbones, such as Deep Shape Matching [6]. They have high image retrieval precisions while retaining clean and compact network designs. However, the lack of global information makes them less accurate when retrieving architectural images from a sketch. Doodle to Search [7] is one of the latest deep learning models for SBIR tasks, where external semantic knowledge is embedded to help domain transformation. It utilises complicated deep neural networks and yields remarkable results, but is resource-intensive due to its sophisticated model designs.

In this paper, we propose a novel compact deep learning model for a practical SBIR task in the scenario of architecture images. The model exploits a siamese network with triplet loss and a view angle classifier. The backbone encoder for the siamese network is a compact vision transformer. Siamese network has long been proven to be an excellent deep learning method for metric learning [8]. The simple structure and the weight sharing mechanism of the siamese network allows us to make a compact SBIR model for practical use. As for the encoder, we choose to use a transformer instead of a CNN because transformers are capable of extracting better global features than CNN models.

Our contribution towards this model can be summarised into two points which the existing SBIR models have rarely considered: 1) improve the model's ability to capture global features by utilising a vision transformer; 2) view angle information of the architecture image is fed to the model to further increase the over precision.

2. Related works

Doodle to Search [7] is a sophisticated deep learning model published recently targeting zero-shot SBIR tasks. This model has two CNN encoders with attention mechanism for processing sketches and images, respectively. An external semantic input is introduced to assist the domain transfer from sketches to images. However, the trade-off for having a complicated network is that the training process is resource-intensive, making it difficult to apply in practical use cases. Deep Shape Matching [6] describes a simple and efficient deep learning model for SBIR tasks. It features an end-to-end process for training an SBIR model without having a single sketch in the dataset. An edge extractor is utilised to generate imitated edge maps in the training stage.

Transformer [9] has recently outperformed CNN in computer vision tasks. They exploit a self-attention mechanism that allows them to outperform CNN models at global feature

extraction [10]. A downside of using a transformer is that the calculation of self-attention is resource-heavy. The Swin transformer [11] is a variation of vision transformers. It reduces the computational complexity of the self-attention calculation step using a shifted-window mechanism. SwinTiny is a compact version of the Swin transformer. Replacing the CNN encoder with SwinTiny would likely increase the precision of the siamese model on SBIR tasks.

3. Methodology

The proposed model is a siamese network with a view angle classifier, as shown in Fig. 2. When a training sample triplet is passed to the model, three inputs (anchor, positive example, and negative example) are each passed to an encoder branch of the siamese network, while the positive and negative examples are also passed to the classifier to determine their view angle tags. In each encoder branch, the output embedding of the SwinTiny transformer is concatenated with the view angle tag and passed to a multilayer perceptron (MLP) network a final embedding. This embedding is then passed to a triplet loss function for loss calculation. Note that, during training, the view angle tag for the anchor is obtained by duplicating the view angle tag for the positive example. This is because the sketch used for the anchor is always a perfect match with the positive example in every training sample.

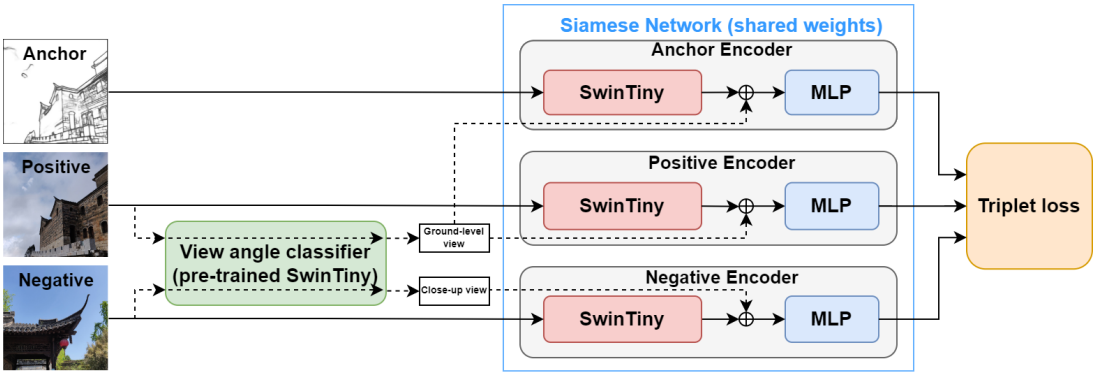

Figure 2. Network structure of the proposed model. The model consists of a siamese network with triplet loss and a pre-trained view angle classifier.

3.1. Siamese network

The siamese network in the proposed model has three branches. Each encoder branch has two inputs: 1) an image x from the training sample triplet, and 2) the view angle x_v of the input image. Before being passed to an encoder, input images are converted from RBG format to grayscale format The conversion is done by taking an average of the RGB values for each pixel. The view angle is obtained by $x_v = C(x)$ where C is the pre-trained view angle classifier described in section 3.2. The complete encoder is denoted by

$$f(x, x_v) = \phi(concat(\sigma(x), x_v)), \tag{1}$$

where σ is a SwinTiny transformer network, and ϕ is an MLP network. The output embedding of the encoder f is a vector of size 256. The triplet loss function $\mathcal{L}$ is defined by

$$\mathcal{L} = max(\delta_p - \delta_n + \alpha, 0). \tag{2}$$

α is a margin parameter for the loss, δ_p and δ_n are the euclidean distances between the embeddings, such that $\delta_p = \|f(x_a, x_{a_v}) - f(x_p, x_{p_v})\|^2$ and $\delta_n = \|f(x_a, x_{a_v}) - f(x_n, x_{n_v})\|^2$. $\{x_a, x_p, x_n\}$ is a triplet denoting the anchor, the positive example, and the negative example.

Figure 3. Examples of images (collected from ArchDaily [4]) in five different view angles: (left to right) top-down view, bird's-eye view, ground-level view, indoor view, close-up view.

3.2. View angle classifier

Adding view angle information to the model helps the encoder create embeddings that better represent the input building designs. We train a classifier to categorise the input building image into one of five view angle classes (Fig. 3). The classifier is a SwinTiny transformer concatenated to an MLP network. The output is a view angle class vector of length five.

4. Experiment

To evaluate the proposed model, we create a sketch-to-image dataset collected from the internet. Following [6], we use an edge extraction model to generate imitated sketches from a set of building images. Each training sample is a triplet containing one sketch and two photos, where the two photos are positive and negative matches of the sketch. One triplet training sample is shown on the left of Fig. 2. We use a pre-trained DexiNed model for edge extraction. It removes some textural details while leaving most structural information in the output sketch [12]. We train three baseline models with the same dataset and compare them against the proposed model.

4.1. Experimental setup

The dataset used for training and testing contains 6,105 building images obtained from ArchDaily [4], each image has a matching sketch generated by a pre-trained DexiNed model. 5,400 out of the 6,105 sketch-image pairs are used for training, and the other 705 pairs are used for testing. The qualitative results in section 4.3 are obtained by running the trained model on a separate building image pool with 34,226 images collected from ArchDaily.

The process for preparing the training sample triplets is broken into the following five steps. 1) Let the sketch-image pool be U, pick a sketch-image pair $(u_s, u_r) \in U$, where u_s is the sketch and u_r is the real image. 2) Calculate the euclidean distances d between the HOG of u_r and every $u_{r_i} \in U_r$. 3) Let a distance threshold be θ, randomly select an image u_{r_i} where $d(u_r, u_{r_i}) > \theta$. Then create a complete training sample $x = \{x_a, x_p, x_n\}$, where $x_a = u_s$, $x_p = u_r$, and $x_n = u_{r_i}$. 4) Repeat step 3 for 40 times, creating 40 training samples for (u_s, u_r). 5) Repeat step 1 to step 4 for every $u_i \in U$, creating $5,400 \times 40$ training samples in total. Unlike the training samples where each sample is a triplet, a test sample is a 2-tuple that contains a sketch and an image. Every test sample is created by matching each sketch with every image in the test data.

In order to assess the performance of the proposed model, we calculate the top-n accuracy using the test results. When processing a test sample, we calculate the Euclidean distance between the sketch and the feature embeddings extracted with a trained encoder. Results with the same sketch are grouped together and ranked by the calculated distances. The top-n accuracy is the average possibility of finding the original image of the sketch in the top-n results.

4.2. Baselines

For comparison, we tested the following three baseline encoders with the same siamese network: ResNet50, SwinTiny with independent anchor branch, and SwinTiny without classifier. ResNet

is a classic network structure in many computer vision tasks [13]. ResNet50 is a compact version of ResNet, and it has a similar number of parameters with the SwinTiny transformer, making it an ideal comparison. All branches in a siamese network share the same set of parameters. Isolating the parameters in the anchor branch provides more freedom to the anchor encoder. This strategy could improve the overall precision since the inputs for the anchor branch are sketches, which are in a different domain than the image inputs for the positive and negative branches. In order to quantify the effect of the view angle classifier on the model accuracy, we also test the encoder without the inputs from the view angle classifier.

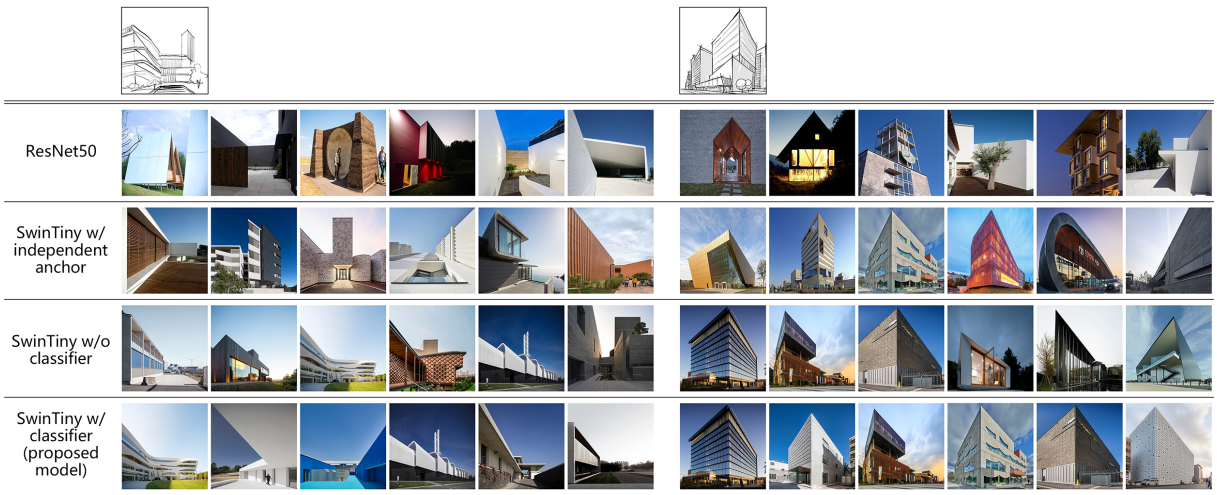

Figure 4. Result comparison between different encoders: top-six matches for two real hand-drawn sketches [3], tested in a separate building image pool.

4.3. Qualitative results

Fig. 4 shows two real hand-drawn sketches and their top-six results produced by each encoder. The result images for the ResNet50 encoder have some resemblance to the sketched building. Some details in the sketch can be recognised in the resulting image. However, the overall structures of the buildings in the result do not match that of the sketch. The SwinTiny encoder with an independent anchor branch produces better results than the previous encoder does, but is still worse than the last two encoders. The reason could be that isolating the anchor parameters causes the model to overfit. The SwinTiny encoder with and without a classifier produces better results than the previous two encoders in terms of global and local feature matching. The structures of the buildings match the sketch to a large degree, reinforcing the assumption that Swin Transformer is effective at capturing global features. The overall similarity of the result images improves slightly with the extra view angle information from the classifier.

4.4. Quantitative results

Table 1 shows the top-n accuracy of the proposed model (last line) and the accuracy of the three baselines. The proposed model reaches 0.859, 0.947, and 0.972 for top-1, top-3, and top-5 accuracy, which is the highest compared to the results of the baseline. The accuracy of the ResNet50 encoder is lower than the second-worst baseline by a large margin. The encoder's accuracy with an independent anchor branch is the second-lowest, despite having double the number of parameters than other encoders. A possible explanation for this observation is that the model over-fits the training samples since the weights of the anchor encoder branch are not paired with the weights of the other two encoder branches. SwinTiny encoder without view angle

classifier inputs performs only slightly worse than the one with view angle inputs, indicating that, to a certain degree, Swin Transformer automatically learns the view angle information.

Table 1. Top-n accuracy and of the siamese network with different backbone encoders.

Encoder	Parameters	Acc.@1	Acc.@3	Acc.@5
ResNet50	24M	0.241	0.428	0.547
SwinTiny w/ independent anchor branch	58M	0.730	0.884	0.919
SwinTiny w/o classifier	29M	0.830	0.931	0.960
SwinTiny w/ classifier (proposed model)	29M	**0.862**	**0.943**	**0.969**

5. Conclusion

We design a siamese network with a Swin Transformer as the backbone encoder. Besides the Swin Transformer encoder, the model also applies a pre-trained Swin Transformer classifier which produces additional information to be sent to the siamese network. Compared to the three baseline models, the proposed model achieves the highest accuracy on a new building image dataset collected from the internet.

6. Acknowledgement

The work was partially supported by the following: National Natural Science Foundation of China under no.61876155; Jiangsu Science and Technology Programme (Natural Science Foundation of Jiangsu Province) under no.BE2020006-4; Key Program Special Fund in XJTLU under no.KSF-T-06.

References

[1] Eitz M, Hildebrand K, Boubekeur T and Alexa M 2011 Sketch-based image retrieval: Benchmark and bag-of-features descriptors vol 17 pp 1624–1636
[2] Hoi C H, Chan C H, Huang K, Lyu M and King I 2004 Biased support vector machine for relevance feedback in image retrieval *IEEE International Joint Conference on Neural Networks* vol 4 pp 3189–3194
[3] 16pic https://www.16pic.com/
[4] Archdaily https://www.archdaily.com/
[5] Huang K, Hussain A, Wang Q and Zhang R 2019 *Deep Learning: Fundamentals, Theory and Applications* Cognitive computation trends (Springer) ISBN 9783030060749
[6] Radenovic F, Tolias G and Chum O 2018 Deep shape matching *European Conference on Computer Vision* pp 774–791
[7] Dey S, Riba P, Dutta A, Llads J L and Song Y Z 2019 Doodle to search: Practical zero-shot sketch-based image retrieval *IEEE Conference on Computer Vision and Pattern Recognition* pp 2174–2183
[8] Hoffer E and Ailon N 2015 Deep metric learning using triplet network *Similarity-Based Pattern Recognition* ed Feragen A, Pelillo M and Loog M pp 84–92
[9] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L u and Polosukhin I 2017 Attention is all you need *Advances in Neural Information Processing Systems* ed Guyon I, Luxburg U V, Bengio S, Wallach H, Fergus R, Vishwanathan S and Garnett R p 59986008
[10] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J and Houlsby N 2021 An image is worth 16x16 words: Transformers for image recognition at scale *International Conference on Learning Representations*
[11] Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S and Guo B 2021 Swin transformer: Hierarchical vision transformer using shifted windows *IEEE International Conference on Computer Vision* pp 10012–10022
[12] Soria X, Riba E and Sappa A 2020 Dense extreme inception network: Towards a robust cnn model for edge detection *IEEE Winter Conference on Applications of Computer Vision* pp 1912–1921
[13] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *IEEE Conference on Computer Vision and Pattern Recognition* pp 770–778